第3章 プログラム起動までの処理 / 演算時のCPU状態 / 分岐処理 / 周辺機器への書き込み

64ビットArm仮想ボードでC言語プログラムの挙動を見る

土屋 健

実験の概要

C言語のプログラムがCPU上でどのように実行されているかを，QEMUを使って見てみます（**図1**）．

● 本章でトライすること

ここではQEMU環境と実験用のプログラムを準備し，以下を見ていきます．

1）プログラム起動までの処理確認
2）メモリの状態確認
3）演算実行時のCPUの状態確認
4）分岐処理が行われる際のCPUの状態確認
5）周辺機器（UART）へのデータ書き込み

● 仮想環境QEMU

QEMUでは，多数のCPUや周辺デバイスのエミュレーション環境がサポートされています．今回はその中から，64ビットArmアーキテクチャの仮想ボード・タイプ（virt）を使ってみます．

```
https://qemu.readthedocs.io/en/latest/system/arm/virt.html
```

virtデバイスは，実際のハードウェアは存在しない仮想的なArmボードとなっています．参考までにQEMUでサポートされているデバイスは以下のコマンドで確認できます．

```
qemu-system-aarch64 -machine help
```

● 実験のプログラム

実験のために用意するプログラムは2つです．

- C言語の本体プログラム（test.c）
- スタートアップ・プログラム（start.S）

実験の本体プログラムはC言語で実装しています．今回はOSのない組み込みボードでの動作を想定しており，標準ライブラリは使用できないため，そこだけは注意が必要です．実験のためのプログラムなので，分岐処理時の挙動が分かりやすいように，意味のない比較処理なども入っています．

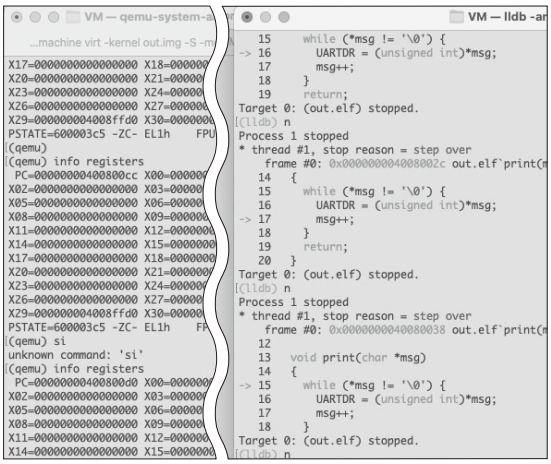

図1　仮想環境QEMUを利用してC言語プログラムの挙動を見てみる

▶C言語の本体プログラム（リスト1）

本体のmain関数（22～49行目）では，加算演算，条件分岐，メッセージ出力を行う処理を実装しています．具体的には後ほど動作確認します．

24～25行目でメッセージ文字列を定義しています．ここではメッセージそのものは初期化データ・セグメントに配置され，その領域を指すポインタ変数がスタックに確保されます．

31行目は加算演算です．

34行目のif文は後続の条件分岐の実験をするために，ステータス・レジスタの条件フラグをNにするためのものです．これがないとここに到達するまでに条件フラグにZが立った状態となっており，後続の動作確認ができないので入れています．

39行目のif文は実装上は必ず真になる処理でENDラベルに飛びます．

47行目のprint関数で指定文字列をシリアル・コンソールに出力して処理は終了となります．

print関数（13～20行目）では，PL011 UARTデバイスに対して文字列を1文字ずつループして書き出している処理です．これでコンソールに文字が表示されます．

本章の開発環境の準備に関してサポート・ページを用意しました．
https://interface.cqpub.co.jp/2207qemu/